I claim:

1. A microprocessor comprising:

a plurality of general purpose registers that are accessible by instructions executing in a non-privileged state;

a plurality of privileged state registers that are accessible by instructions executing in a privileged state; and

a mask register that is writable only when the microprocessor is in said privileged state, said mask register specifying ones of said plurality of privileged state registers to be accessible by instructions executing in a non-privileged state.

2. The microprocessor as recited in claim 1 wherein said non-privileged state comprises a user mode of the microprocessor.

3. The microprocessor as recited in claim 1 wherein said privileged state comprises a kernel mode of the microprocessor.

4. The microprocessor as recited in claim 1 wherein said plurality of privileged state registers comprises coprocessor zero (CP0) registers within a MIPS architecture microprocessor.

5. The microprocessor as recited in claim 1 wherein said plurality of privileged state registers comprise:

a count register that increments at a constant rate; and

a performance counter register for counting events or cycles under a specified set of conditions.

6. The microprocessor as recited in claim 5 wherein said specified set of conditions are specified by a control register for said performance counter register.

7. The microprocessor as recited in claim 1 wherein said mask register comprises:

a plurality of mask bits, each of said plurality of mask bits corresponding to one of said plurality of privileged state registers.

8. The microprocessor as recited in claim 7 wherein when ones of said plurality of mask bits are set, said corresponding ones of said plurality of privileged state registers are made accessible to instructions executing in said non-privileged state.

9. The microprocessor as recited in claim 7 wherein when ones of said plurality of mask bits are not set, said corresponding ones of said plurality of privileged state registers are made inaccessible to instructions executing in said non-privileged state.

10. The microprocessor as recited in claim 1 wherein said ones of said privileged state registers that are accessible by instructions executing in a non-privileged state are read-only.

11. A method for providing user mode access to specified privileged state registers within a microprocessor, comprising:

providing a mask register, writable while in a kernel mode of the microprocessor, the mask register containing a plurality of bits corresponding to a plurality of privileged state registers;

setting specified ones of the plurality of bits, while in the kernel mode of the microprocessor; and

if a user mode instruction requests access to one of the plurality of privileged state registers, and if a corresponding one of the plurality of bits within the mask register has been set by said setting, providing the contents of the one of the plurality of privileged state registers to the user mode instruction.

12. The method as recited in claim 11 wherein said setting comprises:

while in the kernel mode of the microprocessor, writing a value of "1" to the specified ones of the plurality of bits within the mask register for those ones of the plurality of privileged state registers that are to be made accessible.

13. The method as recited in claim 11 further comprising:

if a user mode instruction requests access to one of the plurality of privileged state registers, and if a corresponding one of the plurality of bits within the mask register has not been set by said setting, not providing the contents of the one of the plurality of privileged state registers to the user mode instruction.

14. The method as recited in claim 11 further comprising:

if a user mode instruction requests access to one of the plurality of privileged state registers, and if a corresponding one of the plurality of bits within the mask register has not been set by said setting, generating an exception.

15. The method as recited in claim 14 wherein the exception comprises:

in kernel mode, determining whether access to the one of the plurality of privileged state registers is desirable;

if access is desirable, causing the corresponding one of the plurality of bits within the mask register to be set; and

restarting execution of the user mode instruction.

16. A privileged register set within a microprocessor, comprising:

a plurality of privileged registers that are accessible by instructions executing while the microprocessor is in a kernel mode; and

a mask register for containing a bit pattern, said bit pattern specifying ones of said plurality of privileged registers to be read-only accessible by instructions executing while the microprocessor is in a user mode.

17. The privileged register set as recited in claim 16 wherein said mask register comprises a plurality of bits, each of said bits having a one to one correspondence to ones of said plurality of privileged registers.

18. The privileged register set as recited in claim 17 wherein for those of said plurality of bits that are set, their corresponding ones of said plurality of privileged registers are read-only accessible by instructions executing while the microprocessor is in said user mode.

19. The privileged register set as recited in claim 16 wherein said mask register is loaded with said bit pattern by an instruction executing while the microprocessor is in said kernel mode.

20. A computer program product for use with a computing device, the computer program product comprising:

a computer usable medium, having computer readable program code embodied in said medium, for causing a microprocessor to be described, said computer readable program code comprising:

first program code for providing a plurality of general purpose registers that are accessible by instructions executing in a non-privileged state; and

second program code for providing a plurality of privileged state registers that are accessible by instructions executing in a privileged state; and

third program code for providing a mask register that is writable only when the microprocessor is in said privileged state, said mask register specifying ones of said plurality of privileged state registers to be accessible by instructions executing in a non-privileged state.

21. The computer program product group as recited in claim 20 wherein said plurality of privileged state registers comprise:

a count register that increments at a constant rate; and

a performance counter register for counting events or cycles under a specified set of conditions.

22. The computer program product group as recited in claim 20 wherein said mask register comprises:

a plurality of mask bits, each of said plurality of mask bits corresponding to one of said plurality of privileged state registers.

23. A computer data signal embodied in a transmission medium comprising:

computer-readable program code for providing a privileged register set within a microprocessor, said program code comprising:

first program code for providing a plurality of privileged registers that are accessible by instructions executing while the microprocessor is in a kernel mode; and

second program code for providing a mask register for containing a bit pattern, said bit pattern specifying ones of said plurality of privileged registers to be read-only accessible by instructions executing while the microprocessor is in a user mode.

24. The computer data signal as recited in claim 23 wherein said bit pattern comprises a plurality of bits having a one to one correspondence to ones of said plurality of privileged registers.

25. The computer data signal as recited in claim 24 wherein for those of said plurality of bits that are set, their corresponding ones of said plurality of privileged registers are read-only accessible by instructions executing while the microprocessor is in said user mode.

26. A method allowing an operating system to control user mode access to privileged architecture registers within a microprocessor, the method comprising:

providing a mask register within the microprocessor, the mask register having bits that correspond to the privileged architecture registers;

within the operating system, setting particular bits within the mask register, the particular bits corresponding to ones of the privileged architecture registers that are to be made accessible to a request instruction executing when the microprocessor is in user mode;

upon execution of the request instruction that requests access to one of the privileged architecture registers, determining whether a bit within the mask register, corresponding to the requested one of the privileged architecture registers, has been set; and

if the bit has been set, providing access to the requested one of the privileged architecture registers.

27. The method as recited in claim 26 further comprising:

if the bit has not been set, not providing access to the requested one of the privileged architecture registers.

28. The method as recited in claim 26 wherein when a privileged architecture register has been made accessible, the privileged architecture register is made read-only to the request instruction.

29. The method as recited in claim 26 wherein the mask register is written to when the microprocessor is operating in kernel mode.